In the Specification

Please amend the specification of this application as follows:

Rewrite the paragraph at page 11, line 20 to page 12, line 8 as follows:

B2

--Also illustrated in Figure 2b is an additional storage device, which for reasons more clear below is referred to in this document as an annul word. In the preferred embodiment an annul word includes binary elements corresponding to instructions in a given instruction group. For example, in the illustration of Figure 2b, an annul word AW1 is shown with N elements identified $E1_1$ through $E1_N$, where the "1" in the E1 indicator is used to associate the element with annul word AW1 and the subscript simply increments from one end of the word to the other. Each of elements $E1_1$ through $E1_N$ corresponds to an instruction in group G2 along the same row in Figure 2b. Note also in this regard that a condition is considered a type of instruction and, therefore, each condition also has an associated element in annul word AW1. In all events, as examples of this alignment element $E1_0$ corresponds to instruction 1, element $E1_1$ corresponds to instruction 2, element $E1_2$ corresponds to instruction (and condition) B2, and element $E1_{11}$ corresponds to instruction 10. Further and as explained later, note that no element corresponds to the instruction at the top of group G2 (i.e., instruction ~~A1~~ A2). Finally, given the discussion thus far and the additional detail in the remainder of this document, one skilled in the art will appreciate that an annul word may be performed using various storage or state devices, where a register may be used by way of ~~a~~ an example. --

Rewrite the paragraph at page 17, lines 9 to 26 as follows:

B3

--Figure 5b illustrates instruction group ~~G2~~ G3 along with its corresponding annul word as set by method 20 when condition A3 is

B3 Cont'd.

assumed to be true. More particularly, when method 20 operates with respect to group ~~G2~~ G3, step 24 detects condition A3, and since it is not annulled then step 26 passes the flow to step 28 which advances the flow to step 12 because instruction A3 is not a bottom level instruction. Step 12 assumes that condition A3 is true (in the example of Figure 5b), and next therefore step 14 sets each element in annul word AW2 corresponding to the "ELSE" path of group ~~G2~~ G3. Thus, with respect to Figure 5a, step 14 sets each element in annul word AW2 corresponding to an instruction along the path below and to the right of condition A3, and from Figure 5b it will be shown that these set elements correspond to instructions 2, C3, 5, 6, F3, G3, 11, 12, 13, 14, 19, 20, 21, and 22. These set bits will eventually cause these instructions to be annulled, and note that the annulled instructions include three conditions (i.e., C3, F3, and G3). Since these conditions are annulled then there is no additional processor burden required for testing these conditions and acting in response to the test. Lastly, Figure 5c illustrates instruction group ~~G2~~ G3 along with its corresponding annul word as set by method 20 when condition A3 is assumed to be false and, from the preceding, one skilled in the art should appreciate that its annul word states are complementary to those shown in Figure 5b, thereby annulling the instructions along the path below and to the left of condition A3.--

---

Rewrite the paragraph at page 17, line 27 to page 18, line 15 as follows:

B4

--Figure 5d illustrates instruction group ~~G2~~ G3 along with its corresponding annul word as set by method 20 when condition A3 is assumed to be true and condition B3 is also assumed to be true. Specifically, recall that method 20, after a first instance of either step 14 or 16, returns to process the next condition. Thus, for the example of Figure 5d when condition A3 is first assumed to

be true, then the next instance of step 24 detects condition B3. Next, step 26 determines whether condition B3 is annulled, and this determination is made by referring to annul word AW2 of Figure 5b which corresponds to the present example where condition A3 is assumed to be true; from this analysis, it is determined that condition B3 is not annulled and, thus, step 26 passes the flow to step 28. Since condition B3 is not a bottom level condition, then step 28 passes the flow to step 12. In the example of Figure 5b, condition B3 is also assumed to be true, so the operation of step 12 passes the flow again to step 14, which now sets the bits in annul word AW2 with respect to the instructions below condition B3. Thus, in addition to those bits that were set earlier when condition A3 was processed, and given that step 14 is now operating with respect to condition B3 which is assumed to be true, then step 14 sets the bits in annul word AW2 corresponding to those instructions along the path below and to the right of condition B3. Accordingly, Figure 5d illustrates that, in addition to the same set bits as shown for condition A3 being true from Figure 5b, the bits corresponding to instructions 4, E3, 9, 10, 17, and 18 are also set.--

Rewrite the paragraph at page 19, line 18 to page 20, line 13 as follows:

--Turning now to the details of system 40, it includes an annul word, which to distinguish from earlier examples is designated as AW3. In the preferred embodiment, annul word AW3 includes a sufficient number of bit elements to accommodate a desirable number of instructions in the instruction sequence. For example, under current architectures, a desirable size may be 32 instructions and, thus, as shown in Figure 6a, annul word AW3 includes 32 bit elements $E3_1$ through $E3_{32}$. The states of annul word AW3 are connected to a first data input of a buffer 42 which has a second

input connected to the output of a shift register 44. Shift register 44 has a shift capability equal in size to the number of bits stored by annul word AW3 and, hence, in the current example, shift register 44 is a 32-bit shift register. The shift control input of shift register 44 receives a control signal designated NIP, which is an abbreviation for number of instructions in packet, as will be detailed later. The output of buffer 42 provides a bit group 46. In this regard, note that bit group 46 is referred to in this manner so as to indicate that certain bits, as detailed below, are used from the output of buffer 42, but these bits need not be stored in yet another separate device which could otherwise complicate the design and add delay. The number of bits in bit group 46 is the same number of bits which are in annul word AW3 (i.e., 32 in the present case). The 32 bits of bit group 46 are coupled in a manner such that they are fed back to ~~a second~~ an input of shift register 44, and also such that an integer number M of the least significant of these 32 bits are used as an annul mask AM. Annul mask AM also represents a group of bits rather than a separate hardware device, but for sake of illustration these M bits are shown in Figure 6a as AM_1 through AM_M. Finally, the bits of annul mask AM are coupled to a functional unit use map 48 which, as detailed below, is written by the compiler to map one or more bits from annul mask AM to the appropriate one of the eight functional units FU_1 through FU_8.--

B5 Cont'd.